

REMARKS

Claims 1, 3, 5-16, 18-30, and 37-40 are pending in this application, of which claims 1, 14-16, and 37-40 are independent. Applicant amends claim 14 herein. No new matter is added. Applicant respectfully submits that all of the pending claims are in condition for allowance. Applicant respectfully requests reconsideration of the outstanding rejections and allowance of all pending claims in view of the reasons set forth below.

I. Claim Rejection under 35 U.S.C. § 103(a)

In the Office Action, claims 1, 3, 5-16, 18-30, and 37-40 were rejected under 35 U.S.C. § 103(a) as being obvious over Conway (“Parsing with C++ deferred expressions,” ACM SIGPLAN Notices, vol. 29, no. 9, pp. 9-16, ACM, 1994) (hereafter “Conway”) in view of Walter et al. (“boost C++ Libraries uBLAS overview”) (hereafter “Walter”). Applicant respectfully traverses the rejection.

A. Claim 1

Applicant’s claim 1 recites:

1. In a program development environment, a computer implemented method comprising:
 - providing, via a programming language, a language processor with built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including methods that retrieve values for base language objects;
 - defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;
 - defining said assignment function in more than one class;
 - overloading said assignment function based on a context of said base language objects***; and
 - calling said assignment function from said language processor to determine a value of at least one assignment within at least one of said base language and a base language extension to said base language.

Independent claim 1 recites the feature of ***overloading said assignment function based on a context of said base language objects***. Applicant respectfully submits that Conway and Walter, alone or in any reasonable combination, fail to disclose or suggest this feature of claim

1. The Examiner recognizes that Conway does not disclose or suggest overloading an assignment function based on a context of base language objects (Office Action at page 3).

Instead, the Examiner suggests that Walter discloses “evaluating the left hand side of the overloaded assignment operator” (Office Action at page 3). Applicants respectfully point out that claim 1 does not recite “evaluating the left hand side of the overloaded assignment operator” as alleged by the Examiner. But rather, claim 1 recites ***overloading said assignment function based on a context of said base language objects***. Walter does not disclose ***overloading said assignment function based on a context of said base language objects*** because “evaluating the left hand side of the overloaded assignment operator,” is not the same as ***overloading said assignment function based on a context of said base language objects***, as recited in claim 1.

Further elaborating on the above argument, Walter does not disclose or suggest “context” as use in claim 1. In claim 1, the context of the objects is considered when overloading the assignment function. This allows operations to be carried out more efficiently, or with more user control. Page 2 of the present Application provides the following example. If the assignment “ $y = x + 1$ ” is being evaluated, then an operator overloader may simply add one to x and assign this value to y . If x is a 16-bit variable, then the operation could be carried out in 16 bits. However, if x is a 16-bit variable, but y is only 8-bits wide, the operation ($x+1$) may be carried out in 8 bits. Carrying out the operation in 8-bits is more efficient than a 16-bit implementation, and further is just as accurate as a 16-bit implementation (since the end result is assigned to an 8-bit variable). The method of claim 1 evaluates a *context*, such as the size of the variables involved, in order to allow context-based assignment function overloading. Nothing in the Walter reference indicates that the context of the evaluated objects should be considered.

Conway and Walter, alone or in any reasonable combination, do not disclose or suggest ***overloading said assignment function based on a context of said base language objects***, as present in independent claim 1. Claims 3 and 5-13 depend from claim 1 and, as such, include each and every element of claim 1. Therefore, Conway and Walter, alone or in any reasonable combination, do not disclose or suggest each and every element of claims 3 and 5-13.

Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claims 1, 3, and 5-13.

B. Claim 14

Applicant's amended independent claim 14 recites:

14. In a program development environment, a computer implemented method comprising:

- providing, via a programming language, a language processor with built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including parse tree object methods that retrieve values for base language objects;
- defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;
- defining said assignment function in more than one class;
- overloading said assignment function based on a context of said base language objects, said context determined from one of said parse tree object methods;*
- calling said assignment function from said language processor to determine a value of at least one assignment within at least one of the said base language and a base language extension for said base language; and
- generating code for an embedded processor using said parse tree data structure.

Amended independent claim 14 recites *overloading said assignment function based on a context of said base language objects*. As noted above in the discussion of claim 1, Conway and Walter, alone or in any reasonable combination, do not disclose or suggest *overloading said assignment function based on a context of said base language objects*. Therefore the combination of Conway and Walter do not disclose *overloading said assignment function based on a context of said base language objects* as recited in claim 14..

In order to further clarify the method of claim 14, Applicant amends claim 14 to recite *overloading said assignment function based on a context of said base language objects, said context determined from one of said parse tree object methods*. This amendment clarifies that the parse tree objects provide information about the context of the base language objects. For example, pages 5, 8, and 10 of the present Application describe an example whereby a parse tree object method named “kind” is defined. The “kind” method “tells what kind of an operator is at

the root of the tree” (Application at page 5). At pages 8 and 10, “kind” is called on parse tree object t2 to see if t2 supports in-place operations. This may allow the compiler to use in-place operations at compile-time, which allows compilation to be optimized, “leading to faster, smaller run-time code” (Application at page 9). Conway and Walter, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 14.

Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claim 14.

C. Claim 15

Applicant’s independent claim 15 recites:

15. In a program development environment, a computer implemented method comprising:
- providing, via a programming language, a language processor with built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including methods that retrieve values for base language objects;
 - defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;
 - defining said assignment function in more than one class;
 - overloading said assignment function based on a context of said base language objects;***
 - calling said assignment function from said language processor to determine a value of at least one assignment within at least one of the base language and a base language extension for said base language; and
 - using said parse tree data structure in software emulation.

Independent claim 15 recites ***overloading said assignment function based on a context of said base language***. As noted above in relation to claim 1, Conway and Walter, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of claim 15.

Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claim 15.

D. Claim 16 and 18-30

Applicant's independent claim 16 recites:

16. A computer-readable storage medium for storing computer executable instructions for use in a program development environment, said instructions comprising:

- one or more instructions for providing a programming language;
- one or more instructions for providing a language processor via ~~the~~ said programming language, said language processor having built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including methods that retrieve values for base language objects;
- one or more instructions for defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;
- one or more instructions for defining said assignment function in more than one class;
- one or more instructions for overloading said assignment function based on a context of said base language objects;* and
- one or more instructions for calling said assignment function from said language processor to determine a value of at least one assignment within at least one of said base language and a base language extension to said base language.

Independent claim 16 recites *one or more instructions for overloading said assignment function based on a context of said base language objects*. As noted above in relation to claim 1, Conway and Walter, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of claim 16.

Conway and Walter, alone or in any reasonable combination, do not disclose or suggest each and every element of independent claim 16. Claims 18-30 depend from claim 16 and, as such, include each and every element of claim 16. Therefore, Conway and Walter, alone or in any reasonable combination, do not disclose or suggest each and every element of claims 18-30. Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claims 16 and 18-30.

E. Claim 37

Applicant's independent claim 37 recites:

37. A computer-readable storage medium for storing computer executable instructions for use in a program development environment, said instructions comprising:

- one or more instructions for providing a programming language;
- one or more instructions for providing a language processor via said programming language, said language processor having built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including methods that retrieve values for base language objects;
- one or more instructions for defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;
- one or more instructions for defining said assignment function in more than one class;
- one or more instructions for overloading said assignment function based on a context of said base language objects;***
- one or more instructions for calling said assignment function from said language processor to determine a value of at least one assignment within at least one of said base language and a base language extension to said base language;
- and
- one or more instructions for generating code for an embedded processor using said parse tree data structure.

Independent claim 37 recites ***one or more instructions for overloading said assignment function based on a context of said base language objects***. As noted above in relation to claim 1, Conway and Walter, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of claim 37.

Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claim 37.

F. Claim 38

Applicant's independent claim 38 recites:

38. A computer-readable storage medium for storing computer executable instructions for use in a program development environment, said instructions comprising:

- one or more instructions for providing a programming language;
- one or more instructions for providing a language processor via said programming language, said language processor having built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects,

said parse tree objects including methods that retrieve values for base language objects;

one or more instructions for defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

one or more instructions for defining said assignment function in more than one class;

one or more instructions for overloading said assignment function based on a context of said base language objects;

one or more instructions for calling said assignment function from said language processor to determine a value of at least one assignment within at least one of said base language and a base language extension to said base language; and

one or more instructions for using said parse tree data structure in software emulation.

Independent claim 38 recites ***one or more instructions for overloading said assignment function based on a context of said base language objects***. As noted above in relation to claim 1, Conway and Walter, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of claim 38.

Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claim 38.

G. Claim 39

Applicant's independent claim 39 recites:

39. In an object-oriented program development environment having a base language, a computer implemented method comprising:

providing a programming language;

providing a language processor via said programming language, said language processor having built-in support for a parse tree data structure in said object-oriented program development environment, said parse tree data structure used as a basis for at least one parse tree object, said parse tree objects including methods that retrieve values for base language objects;

defining an assignment function taking a plurality of parse tree structures as arguments in more than one class;

calling said assignment function from said language processor to determine a value of at least one assignment within at least one of a base language and a base language extension; and

overloading a mathematical operator with said assignment function based on a context of a plurality of said base language objects.

Independent claim 39 recites *overloading a mathematical operator with said assignment function based on a context of a plurality of said base language objects*. Conway and Walter, alone or in any reasonable combination, do not disclose or suggest the above feature of claim 39.

The Examiner rejects claim 39 “for the same reasons set forth in connection with the rejection of claim 3 above” (Office Action at page 7). In rejecting claim 3, which depends from claim 1, the Examiner relies on the rejection of claim 1. In the rejection of claim 1, the Examiner recognizes that Conway does not disclose *a context of a plurality of said base language objects*. As noted above with respect to claim 1, Walter also does disclose or suggest utilizing the context of the base language objects.

Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claim 39.

H. Claim 40

Applicant’s independent claim 40 recites:

40. A computer implemented method for providing a language processor in a program development environment, said method comprising:
- building a parse tree data structure based on source code with the language processor;
 - instantiating a first parse tree object and a second parse tree object, each parse tree object including methods that retrieve values for objects represented in the source code;
 - evaluating said second parse tree object to obtain a value, said evaluating done based on a context provided by said first parse tree object*; and
 - assigning said value to said first parse tree object.

Independent claim 40 recites *evaluating said second parse tree object to obtain a value, said evaluating done based on a context provided by said first parse tree object*. Conway and Walter, alone or in any reasonable combination, do not disclose or suggest the above feature of claim 40.

The Examiner recognizes that Conway does not disclose *a context provided by said first parse tree object* (Office Action at page 7). Walter also does disclose or suggest utilizing the

context of an object. Further, Walter does not address parse trees, and so Walter does not disclose or suggest *evaluating a second parse tree object to ... based on a context provided by said first parse tree object*.

Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the 35 U.S.C. §103(a) rejection of claim 40.

CONCLUSION

In view of the above, Applicant believes the pending application is in condition for allowance and urges the Examiner to pass all of the pending claims to allowance. Should the Examiner feel that a teleconference would expedite the prosecution of this application, the Examiner is urged to contact the Applicant's attorney at (617) 227-7400.

Please charge any shortage or credit any overpayment of fees to our Deposit Account No. 12-0080, under Order No. MWS-039RCE. In the event that a petition for an extension of time is required to be submitted herewith, and the requisite petition does not accompany this response, the undersigned hereby petitions under 37 C.F.R. §1.136(a) for an extension of time for as many months as are required to render this submission timely. Any fee due is authorized to be charged to the aforementioned Deposit Account.

Dated: December 23, 2008

Respectfully submitted,

Electronic signature: /Kevin J. Canning/
Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant